

## Chapter 5

### Data Product Labels

PDS data product labels are required for describing the contents and format of each individual data product within a data set. PDS data product labels are written in the Object Description Language (ODL). The PDS has chosen to label the wide variety of data products under archival preparation by implementing a standard set of data object definitions, data elements, and standard values for the elements. These data object definitions, data elements, and standard values are defined in the *Planetary Science Data Dictionary* (PSDD). Appendix A of this document provides general descriptions and examples of the use of these data object definitions and data elements for labeling data products.

#### 5.1 Format of PDS Labels

##### 5.1.1 Labeling methods

In order to identify and describe the organization, content, and format of each data product, PDS requires a distinct data product label for each individual data product file. These distinct product labels may be constructed in one of three ways:

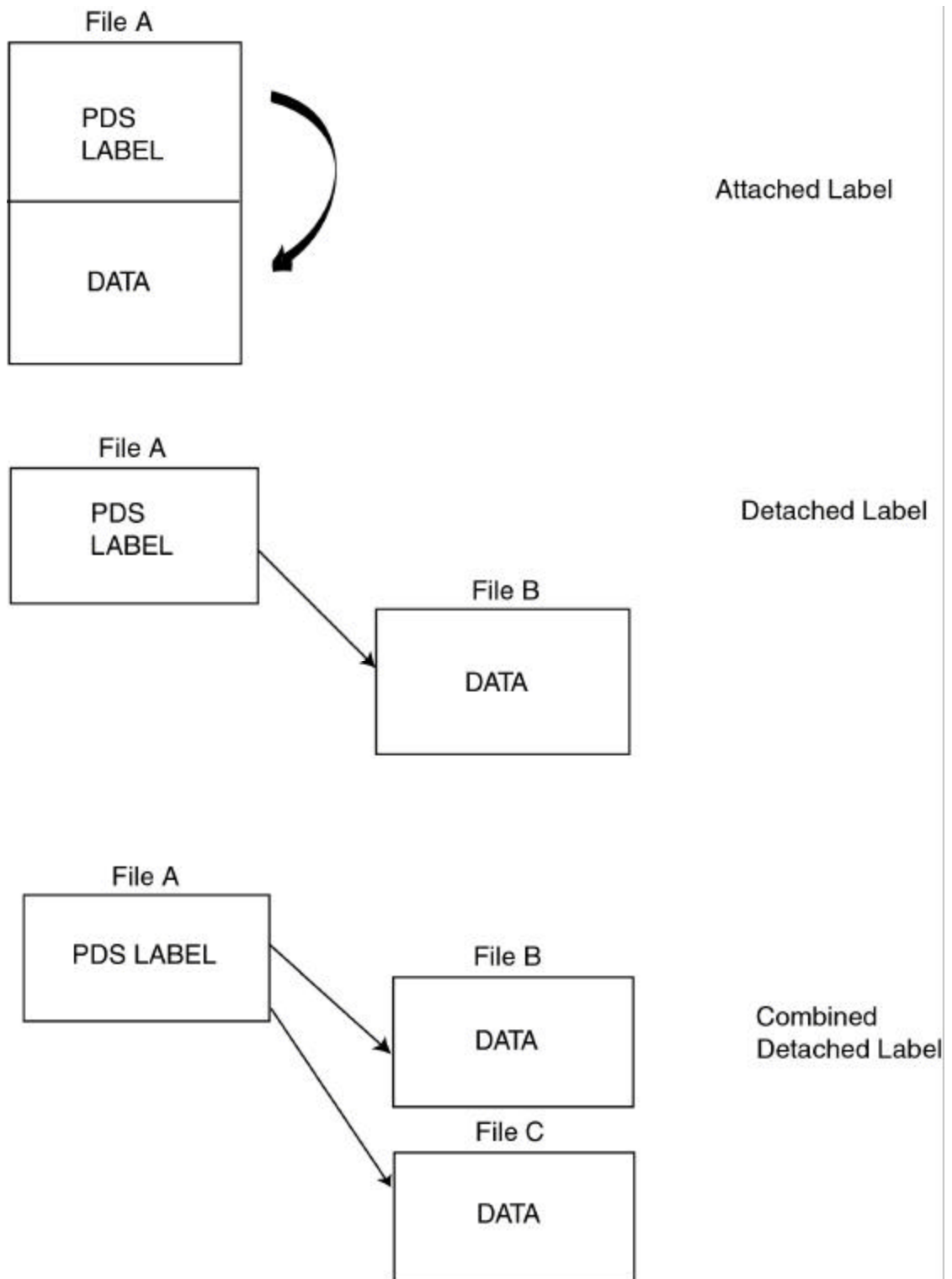
Attached - The PDS data product label is embedded at the beginning of the data product file. There is one label attached to each data product file.

Detached - The PDS data product label is detached from the data and resides in a separate file which points to the data product file. There is one detached label file for every data product file. The label file should have the same base name as its associated data file, but the extension .LBL .

Combined Detached - A single PDS detached data product label file is used to describe the contents of more than one data product file. The combined detached label points to individual data products.

NOTE: Although all three labeling methods are equally acceptable, the PDS tools do not currently support the Combined Detached label option.

Figure 5.1 illustrates the use of each of these methods for labeling individual data product files.



*Figure 5.1 Attached, Detached, and Combined Detached PDS Labels*

### 5.1.2 Label format

PDS recommends that labels have stream record format, and line lengths of at most 80 characters (inclusive of the CR/LF line terminators) so that the entire label can be seen on a computer screen without horizontal scrolling. The Carriage Return and Line Feed (CR/LF) pair is the required line terminators for all PDS labels. (See the *Record Formats* chapter of this document.)

All values in a PDS label should be in upper case, except values for description fields. It is also recommended that the equal signs in the labels be aligned for ease of reading.

#### *ASCII Character Set*

All values in a PDS label must conform to the standard ASCII character set. These values include the range of characters 001 through 127 (decimal). The character set 128-255 is not used in PDS because these characters vary by font / symbol set and are not predictable when viewed and/or printed. Similarly, TAB characters are not to be used as they are interpreted differently by different applications. Each TAB character represents 'n' number of SPACE characters and 'n' will vary by application thereby causing nicely formatted tables/columns to become misaligned.

#### *Label Padding*

For a fixed length data file with an attached label, the label is padded with SPACES (ASCII decimal 32) in one of the following ways:

- 1) Spaces are added after the label's END <CR><LF> statement and before the data so that the total size of the label is an integral multiple of the record length of the data.

Example:

In the example below, the label portion of the file is  $7 \times 324 = 2268$  bytes in length, including blank fill between the END<CR><LF> statement and the first byte of data. The actual data portion of the file starts at record 8 (i.e., the 1st byte of the 8th record starts at byte  $(7 \times 324) + 1 = 2269$ )

```

RECORD_TYPE           = FIXED_LENGTH<CR><LF>
RECORD_BYTES          = 324<CR><LF>
FILE_RECORDS          = 334<CR><LF>
LABEL_RECORDS         = 7<CR><LF>

^IMAGE                = 8<CR><LF>

END<CR><LF>
...blank fill...
data
```

- 2) Each line in the label may be padded with SPACES so that each line in the label has the same record length as the data file. In this case, the label line length may exceed the recommended 80 characters.

Example:

In the example below, the label portion of the file is  $80 \times 85 = 6800$  bytes in length. Each line in the label portion of the file is 85 bytes long, the same length as each data record. Notice the blank space between the actual values in the label and the line delimiters. In the example, the label is 80 lines long (i.e., 80 records long) and the data begins at record 81. Note that the label is padded so that <CR><LF> are in bytes 84 and 85.

RECORD_TYPE	= FIXED_LENGTH	<CR><LF>
RECORD_BYTES	= 85	<CR><LF>
FILE_RECORDS	= 300	<CR><LF>
LABEL_RECORDS	= 80	<CR><LF>
...		
^TABLE	= 81	<CR><LF>
END		<CR><LF>
data		

## 5.2 Data Product Label Content

### 5.2.1 Attached and Detached Labels

PDS data product labels have a general structure that is used for all attached and detached labels, except for data products described by minimal labels. (Minimal labels are described in Section 5.2.3.)

- LABEL STANDARDS identifier
- FILE CHARACTERISTIC data elements
- DATA OBJECT pointers
- IDENTIFICATION data elements
- DESCRIPTIVE data elements
- DATA OBJECT DEFINITIONS
- END statement

Figure 5.2 provides an example of how this general structure appears in an attached or detached label for a data product file containing multiple data objects.

PDS LABEL		
CCSD. . .	/* optional SFDU */	• LABEL STANDARDS IDENTIFIERS
PDS_VERSION_ID	=	
/*FILE_CHARACTERISTICS */		• FILE CHARACTERISTICS DATA ELEMENTS
RECORD_TYPE	=	
RECORD_BYTES	=	
FILE_RECORDS	=	
LABEL_RECORDS	=	
/*POINTERS TO DATA OBJECTS */		• DATA OBJECT POINTERS (primary, secondary)
^IMAGE	=	
^HISTOGRAM	=	
/*IDENTIFICATION DATA ELEMENTS */		• IDENTIFICATION DATA ELEMENTS
DATA_SET_ID	=	
PRODUCT_ID	=	
SPACECRAFT_NAME	=	
INSTRUMENT_NAME	=	
TARGET_NAME	=	
START_TIME	=	
STOP_TIME	=	
.		
.		
.		
PRODUCT_CREATION_TIME	=	
/*DESCRIPTIVE DATA ELEMENTS */		• DESCRIPTIVE DATA ELEMENTS
FILTER_NAME	=	
OFFSET_MODE_ID	=	
.		
.		
.		
/*DATA OBJECT DEFINITIONS */		• DATA OBJECT DEFINITIONS (primary, secondary)
OBJECT	= IMAGE	
.		
.		
END_OBJECT	= IMAGE	
OBJECT	= HISTOGRAM	
.		
.		
END_OBJECT	= HISTOGRAM	
END	CCSD . . .	• END STATEMENT

*Figure 5.2 PDS Attached / Detached Label Structure*

### 5.2.2 Combined Detached Labels

For the Combined Detached label option, the general label structure is modified slightly to explicitly reference each individual file within its own FILE object. In addition, identification and descriptive data elements that apply to all of the files can be located before the FILE objects.

- LABEL STANDARDS identifiers
- IDENTIFICATION data elements that apply to all referenced data files
- DESCRIPTIVE data elements that apply to all referenced data files
- OBJECT=FILE statement (Repeats for each data product file)
  - FILE CHARACTERISTIC data elements
  - DATA OBJECT pointers
  - IDENTIFICATION data elements
  - DESCRIPTIVE data elements
  - DATA OBJECT DEFINITION
- END\_OBJECT=FILE statement
- END statement

Figure 5.3 provides an example of how this general structure appears in a combined detached label that describes more than one data product file.

PDS LABEL		
CCSD...	/* optional SFDU */	• LABEL STANDARDS IDENTIFIERS
PDS VERSION ID	=	
DATA_SET_ID	=	• IDENTIFICATION & DESCRIPTIVE DATA ELEMENTS for all files
PRODUCT_ID	=	
SPACECRAFT_ID	=	
INSTRUMENT_NAME	=	
TARGET_NAME	=	
PRODUCT_CREATION_TIME	=	
OBJECT_FILE		• For Detached FILE A: FILE CHARACTERISTICS DATA ELEMENTS
RECORD_TYPE	=	
.		
.		
.		
FILE_RECORD	=	• DATA OBJECT POINTERS • IDENTIFICATION/DESCRPTIVE DATA ELEMENTS • DATA OBJECT DEFINITIONS
^TIME_SERIES	= "FILEA"	
START_TIME	=	
STOP_TIME	=	
OBJECT	= TIME_SERIES	
.		
.		
.		
END_OBJECT	= TIME_SERIES	
END_OBJECT	= FILE	
OBJECT	= FILE	For Detached FILE B: • FILE CHARACTERISTICS DATA ELEMENTS
RECORD_TYPE	=	
.		
.		
.		
FILE_RECORD	=	• DATA OBJECT POINTERS • IDENTIFICATION/DESCRIPTIVE DATA ELEMENTS • DATA OBJECT DEFINITIONS
^TIME_SERIES	= "FILEB"	
START_TIME	=	
STOP_TIME	=	
OBJECT	= TIME_SERIES	
.		
.		
.		
END_OBJECT	= TIME_SERIES	
END_OBJECT	= FILE	
END	CCSD...	• END STATEMENT

Figure 5.3 PDS Combined / Detached PDS Label Structure

### 5.2.3 Minimal Labels

Use of the minimal label option is only allowed when the format of the data cannot be supported by the current documented Data Objects.

For minimal labels, the general label structure has removed the required use of data objects. A minimal label does not contain any PDS data object definitions or pointers to data objects. The above applies to both attached and detached labels.

Minimal labels must satisfy the following requirements:

- (1) Provide the ability to locate the data (file) associated with the label.

- 1a. Attached labels

Since data objects and pointers are not required in the minimal label, by definition the data follows immediately after the label.

- 1b. Detached Labels

Both the implicit and explicit use of the FILE object are supported. The FILE\_NAME keyword, contained in the FILE object, is required.

- (2) Provide the ability to locate a description of the format/content of the data. One of the following must be provided in the minimal label:

- 2a. ^DESCRIPTION = "<filename>"

This is a pointer to a file containing a detailed description of the data format; may be located in the same directory as the data or in the DOCUMENT subdirectory.

- 2b. DESCRIPTION = "<text appears here>"

This is either a detailed description of the data file, its format, data types, and use, or it is a reference to a document available externally, e.g., a Software Interface Specification (SIS) or similar document.

- (3) When minimal labels are used, DATA\_OBJECT\_TYPE = FILE should be used in the Data Set Catalog template. If a situation arises where multiple Data Object types are used, then separate Data Set Catalog templates should be provided for each data product. And, where appropriate, use a Data Set Collection template.

#### 5.2.3.1 Implicit File Object (Attached and Detached Minimal Label)

The general structure for minimal labels with implicit file objects is as follows:

- LABEL STANDARDS identifier

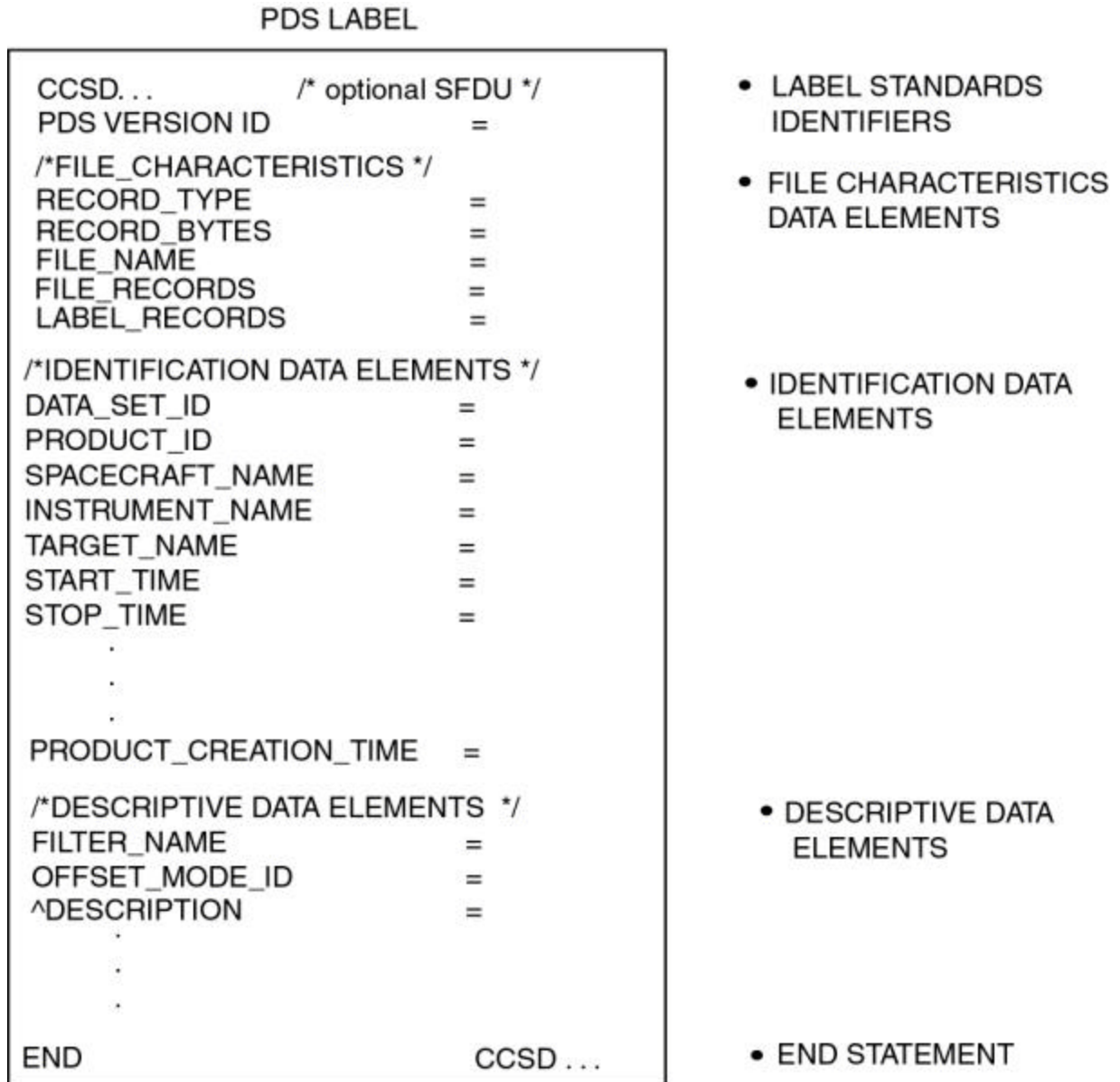
- FILE CHARACTERISTIC data elements
- IDENTIFICATION data elements
- DESCRIPTIVE data elements
- END statement

### 5.2.3.2 Explicit File Object (Detached Minimal Label)

The general structure for minimal labels with explicit file objects is as follows:

- LABEL STANDARDS identifier
- IDENTIFICATION data elements
- DESCRIPTIVE data elements
- OBJECT=FILE statement
  - FILE CHARACTERISTIC data element
- END\_OBJECT=FILE
- END statements

Figure 5.4 provides an example of how this general structure appears in a detached minimal label. In this example, an implicit FILE object is used.



*Figure 5.4 PDS Detached Minimal Label Structure*

### 5.3 Detailed Label Contents Description

This section describes the detailed requirements for the content of PDS labels. The subsections describe label standards identifiers, file characteristic data elements, data object pointers, identification data elements, descriptive data elements, data object definitions, and the END statement.

### 5.3.1 Label Standards Identifiers

Each PDS label begins with an optional Standard Formatted Data Unit (SFDU) label and a PDS\_VERSION\_ID data element:

```
CCSD....      [optional SFDU label]
PDS_VERSION_ID
```

The PDS does not require SFDU labels on individual products, but they may be needed for conformance with specific project or other agency requirements. If SFDUs are provided on a data product, they must follow the standards described in the *SFDU Usage* chapter in this document.

The PDS requires the PDS\_VERSION\_ID data element to identify the PDS published standards that the label adheres to. This version id will be used to provide PDS software tool support for a specific set of standards and will allow the evolution and expansion of both standards and tools as required by the PDS user community.

For labels adhering to the standards described in this document -- the *PDS Standards Reference*, Version 3.3 -- and its associated *Planetary Science Data Dictionary*, Version 3.0, this will be:

```
PDS_VERSION_ID = PDS3
```

### 5.3.2 File Characteristic Data Elements

PDS data product labels contain data element information that describes important attributes of the physical structure of a data product file. The PDS file characteristic data elements are:

```
RECORD_TYPE
RECORD_BYTES
FILE_RECORDS
LABEL_RECORDS
```

The RECORD\_TYPE data element identifies the record characteristics of the data product file. A complete discussion of the RECORD\_TYPE data element and its use in describing data products produced on various platforms is provided in the *Record Formats* chapter in this document. The RECORD\_BYTES data element identifies the number of bytes in each physical record in the data product file. The FILE\_RECORDS data element identifies the number of physical records in the file. The LABEL\_RECORDS identifies the number of physical records that make up the PDS product label.

Not all of these data elements are required in every data product label. Table 5.1 lists the required (Req) and optional (Opt) file characteristic data elements for a variety of data products and labeling methods for both attached (Att) and detached (Det) labels. Where (max) is specified, the value indicates the maximum size of any physical record in the file.

**Table 5.1: File Characteristic Data Element Requirements**

Labeling Method	Att	Det	Att	Det	Att	Det	Att	Det
RECORD_TYPE			FIXED_LENGTH		VARIABLE_LENGTH		STREAM	
RECORD_BYTES	Req	Req	Rmax	Rmax	Omax	-	-	-
FILE_RECORDS	Req	Req	Req	Req	Opt	Opt	-	-
LABEL_RECORDS	Req	-	Req	-	Opt	-	-	-

Note: For detached minimal labels, the FILE\_NAME keyword is required.

### 5.3.3 Data Object Pointers

The actual data whose structure and attributes are defined in a PDS label are “data objects”. Each data product file may contain one or more data objects.

The PDS uses a pointer mechanism within product labels to identify the starting locations for all primary and secondary data objects in a data product.

Examples of primary data objects that may require data object pointers include:

TABLE  
IMAGE  
SERIES  
SPECTRUM  
QUBE

Examples of secondary data objects that may require data object pointers include:

HISTOGRAM  
PALETTE  
HEADER  
DOCUMENT

#### 5.3.3.1 Use of Pointers in Attached Labels

For attached labels, if there is only one data object referenced, a data object pointer is not required. However, it is strongly recommended that data object pointers be used at all times. The data object is assumed to start in the next physical record after the PDS product label area. This is commonly the case with ASCII text files described by a TEXT object and ASCII SPICE files described by a SPICE\_KERNEL object. The top two illustrations in Figure 5.5 show example files that do not require data object pointers.

If multiple data objects are stored in the data product file, object pointers are required for all data

objects. The syntax for data object pointers in attached labels may take one of two forms:

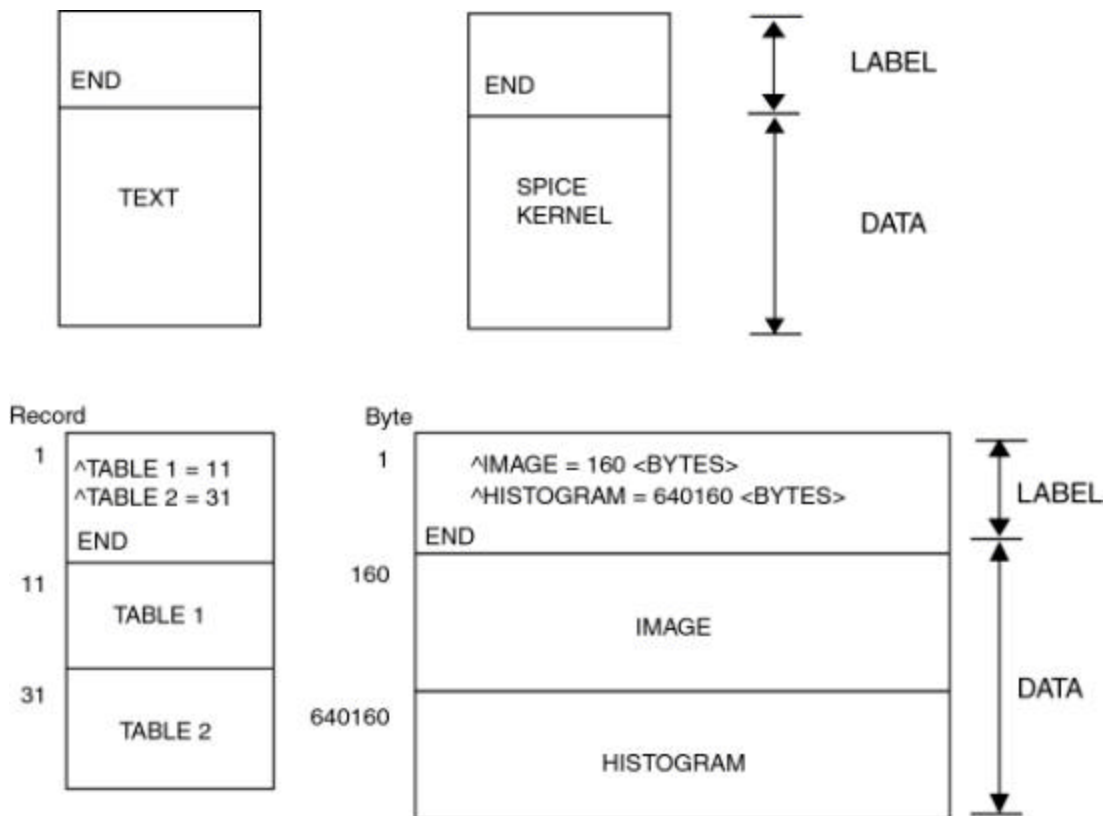
$\wedge\langle\text{object\_identifier}\rangle = \text{nnn}$  (see the *Object Description Language* chapter in this document)

where nnn represents the starting record number within the file,  
Or,

$\wedge\langle\text{object\_identifier}\rangle = \text{nnn} \langle\text{BYTES}\rangle$

where nnn represents the starting byte location within the file.

The following two illustrations in Figure 5.5 show the required use of data object pointers for attached label products containing multiple data objects.



*Figure 5.5 Data Object Pointers-Attached Labels*

### 5.3.3.2 Use of Pointers in Detached and Combined Detached Labels

If the PDS data product label is a detached or a combined detached label, data object pointers are required for all data objects referenced.

The syntax for data object pointers may take one of three forms:

- (1) *^object\_identifier* = "filename"
- (2) *^object\_identifier* = ("filename", nnn)
- (3) *^object\_identifier* = ("filename", nnn <BYTES>)

With respect to these three cases, these object pointers reference either byte or record locations in data files that are detached, or separate from, the label file.

In each case, the filename is the name of the file containing the data object and is detached, or separate from, the label file. In the first case, the data object is located at the beginning of the referenced file. In the second case, the data object begins nnn physical records from the beginning of the referenced file. In the third case, the data object begins nnn bytes from the beginning of the referenced file.

```

^QUBE                = ("DATA.DAT")
^ENGINEERING_TABLE   = ("DATA.DAT", 10)
^TABLE                = ("DATA.DAT", 10 <BYTES>)

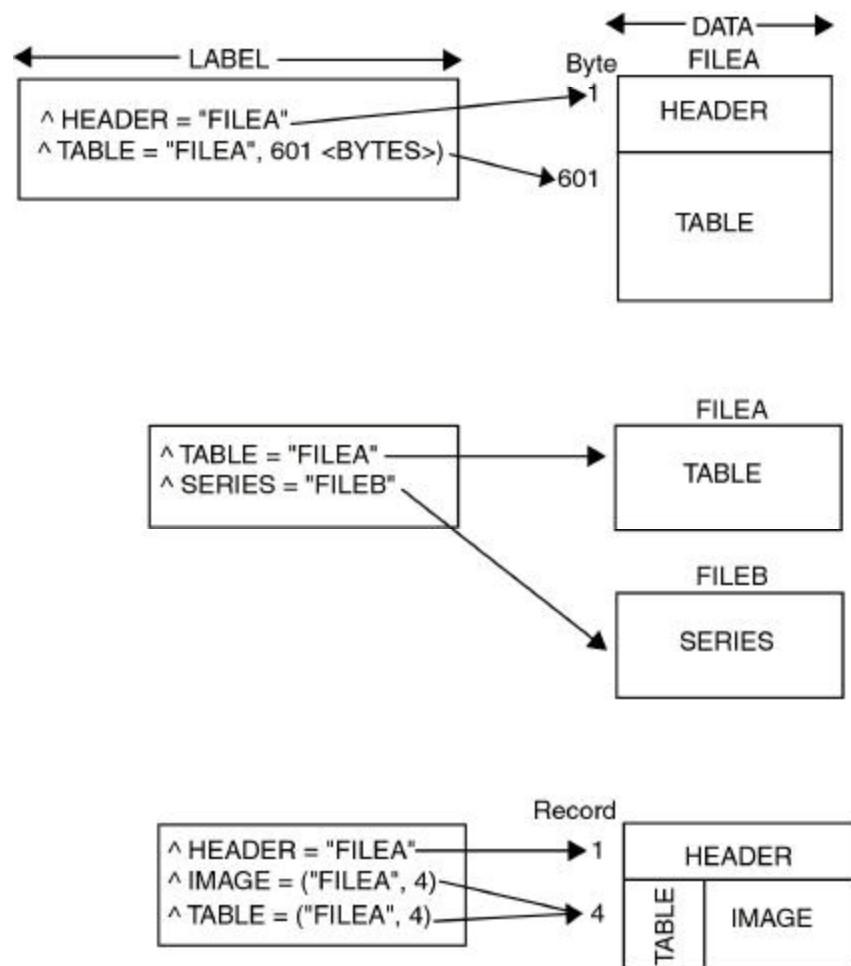
```

Figure 5.6 illustrates several examples of data object pointer usage for data product files with detached or combined detached labels. The top example shows a data product consisting of a HEADER data object and a TABLE data object together in a single file. The detached label for this product includes pointers for both data objects, with the TABLE object starting at byte 601 of file A. The middle example illustrates a combined detached label for a data product contained in two data objects, each in a separate file. A separate pointer is provided for each data object. The bottom example shows a detached label for a data product containing multiple data objects.

Where multiple data objects are stored within a data product file, and where multiple data objects occupy portions of the same physical record, the data object pointer indicates the first physical record containing the data object. Additional data elements within the Data Object Definitions (e.g. LINE\_PREFIX\_BYTES, ROW\_SUFFIX\_BYTES) provide the relative byte locations within each record for each line or row of data within the data object.

### 5.3.3.3 Note Concerning Minimal Attached and Detached Labels

By definition, data object pointers do not exist in minimal labels. The format of the data is fully described in a separate file or document.



**Figure 5.6 Data Object Pointers – Detached & Combined Labels**

### 5.3.4 Identification Data Elements

The identification data elements provide important information about the data to uniquely identify the data product and to associate it with other data products that may be related. This information is often used to populate the PDS product level catalogs or inventories. PDS requires a minimum set of these identification data elements to be included in all product labels. These requirements vary depending on the type of data product being archived. Additional identifying data elements may be required by specific projects or organizations.

Additional data elements which might be needed to further identify the data objects or which would be needed to catalog the data product to support potential search criteria should also be included. These additional data elements are selected from the *Planetary Science Data Dictionary* (PSDD).

**NOTE:** When a data element is needed for a data product label, but is not yet recorded in the PSDD, it can be proposed to be added to the dictionary. See a PDS Data Engineer for assistance.

### 5.3.4.1 Spacecraft Science Data Products

The following identification data elements shall be included in data product labels for all spacecraft science data products:

DATA\_SET\_ID  
PRODUCT\_ID  
INSTRUMENT\_HOST\_NAME  
INSTRUMENT\_NAME  
TARGET\_NAME  
START\_TIME  
STOP\_TIME  
SPACECRAFT\_CLOCK\_START\_COUNT  
SPACECRAFT\_CLOCK\_STOP\_COUNT  
PRODUCT\_CREATION\_TIME

### 5.3.4.2 Earthbased Science Data Products

The following identification data elements shall be included in data product labels for all earthbased science and radio science data products:

DATA\_SET\_ID  
PRODUCT\_ID  
INSTRUMENT\_HOST\_NAME  
INSTRUMENT\_NAME  
TARGET\_NAME  
START\_TIME  
STOP\_TIME  
PRODUCT\_CREATION\_TIME

### 5.3.4.3 Ancillary Data Products

The following identification data elements shall be included in data product labels for all ancillary data sets. These types of products may be more general in nature, supporting a wide variety of instruments for a particular mission. For example, SPICE data sets, general engineering data sets, and uplink data are considered ancillary data products.

DATA\_SET\_ID  
PRODUCT\_ID  
PRODUCT\_CREATION\_TIME

The following data elements are highly recommended, and should be included in ancillary data products whenever they apply:

INSTRUMENT\_HOST\_NAME  
INSTRUMENT\_NAME  
TARGET\_NAME  
START\_TIME  
STOP\_TIME  
SPACECRAFT\_CLOCK\_START\_COUNT  
SPACECRAFT\_CLOCK\_STOP\_COUNT

### 5.3.5 Descriptive Data Elements

In addition to the identification data elements required for various types of data, PDS strongly recommends including additional data elements related to specific types of data. These descriptive data elements must include any data elements which might be needed to interpret or process the data objects or which would be needed to catalog the data product to support potential search criteria at the product level.

Not only will these values be available with the data to the user, but they are also used to load PDS product level catalogs and inventories with descriptive information about each data product. PDS product level catalogs and inventories at PDS Discipline Nodes support both online data product access and ordering capabilities.

In addition, PDS has developed and continues to develop software display and analysis packages for standard data objects. These software packages will be built to utilize various descriptive data elements.

Recommendations for descriptive data elements to consider supplying will come from working with PDS Mission Interface personnel as well as the data producer's own suggestions. These additional data elements are selected from the *Planetary Science Data Dictionary*.

NOTE: When a data element is needed for a data product label, but is not yet recorded in the PSDD, it can be proposed to be added to the dictionary. See the PDS Data Engineer for assistance with submitting new data elements for inclusion in the PSDD.

Pointers are sometimes used in a PDS label to provide a shorthand method for including a set of descriptive data elements (e.g., ^DESCRIPTION) or a long descriptive text passage referenced in several data product labels.

### 5.3.6 Data Object Definitions

The PDS requires a separate data object definition within a product label for describing the structure and associated attributes of each data object in the data product. There will be one data object definition for every primary and secondary data object pointer identified in Section 5.2.3. These data object definitions are of the form:

```
OBJECT          = aaa      where aaa is the name of the data object
...
END_OBJECT      = aaa
```

The PDS has designed a set of standard data object definitions to be used for labeling data products. Among these standard objects are those designed to describe data structures commonly used for scientific data storage. Appendix A provides a complete set of PDS data object definition requirements, along with examples of data product labels.

Pointers are sometimes used in a PDS label to provide a shorthand method for including a set of data sub-objects referenced in several data product labels. For example, a ^STRUCTURE is

often used to include a set of COLUMN sub-objects for a TABLE structure that is used in many labels.

NOTE: Minimal labels do not contain any data object definitions.

### **5.3.7 End Statement**

The end of the PDS label is identified by the required END statement followed by an optional SFDU.

The PDS does not require SFDU labels on individual products, but they may be required to conform with specific project or other agency requirements. If SFDUs are provided on a data product, they must follow the standards described in the *SFDU Usage* chapter in this document. In some, but not all cases, another SFDU label is required after the PDS END statement to provide “end label” and sometimes “start data” information.